

ADAPTIVE BROWSER UNTUK OPTIMALISASI KONSUMSI PENGGUNAAN DAYA BATERAI PADA MOBILE DEVICE

Mohammad Suryawinata¹⁾
suryawinata12@mhs.if.its.ac.id
Institut Teknologi Sepuluh
Nopember

Waskitho Wibisono²⁾
waswib@if.its.ac.id
Institut Teknologi Sepuluh
Nopember

Hudan Studiawan³⁾
hudan@if.its.ac.id
Institut Teknologi Sepuluh
Nopember

ABSTRAK

Berkembangnya teknologi *mobile* akhir-akhir ini mengakibatkan semakin banyak pengguna yang mengakses layanan jaringan melalui perangkat *mobile*, seperti tablet atau *smartphone*. Personalisasi tampilan dan konten menjadi lebih penting. Perangkat *mobile* memiliki sifat yang berbeda dengan komputer *desktop*. Perangkat *mobile* tidak memiliki layar berukuran besar, perangkat input-output yang nyaman, dan memiliki kemampuan komputasi terbatas. Adaptasi konten merupakan definisi umum yang digunakan untuk adaptasi/kustomisasi konten yang ditransfer dari server melalui protokol HTTP ke klien. Metode penelitian yang dilakukan adalah menyesuaikan konten yang akan ditransfer kepada pengguna berdasarkan sumberdaya perangkat yakni baterai pada sisi klien sehingga kualitas layanan yang lebih tinggi akan tercapai. Dalam penelitian ini, peneliti akan melakukan pengembangan metode adaptasi konten yang mempertimbangkan kondisi sumberdaya baterai pada perangkat pengakses. Estimasi sumberdaya baterai dilakukan untuk mengetahui kondisi terkini baterai yang ada pada perangkat pengakses. Mekanisme penentuan konten yang sesuai akan didasarkan pada nilai yang dihasilkan dari estimasi sumberdaya baterai pada perangkat yang telah dilakukan.

Kata Kunci: adaptasi konten, *mobile browser*, *resource aware*, *mobile computing*

I. Pendahuluan

Perangkat *mobile* saat ini merupakan perangkat yang sangat populer dan banyak pengguna yang melakukan aktivitas harian melalui perangkat ini. Pada akhir tahun 2013 jumlah perangkat *mobile* tumbuh menjadi 7 milyar perangkat dengan 77% dari jumlah tersebut adalah *smartphone*[1]. *Smartphone* dan tablet mendominasi jumlah trafik terbesar sejumlah 88% dari total trafik di internet[1].

Website pada awalnya di desain khusus untuk penggunaan komputer *desktop*. *Website* memiliki ukuran (resolusi) sebesar 800 x 600 piksel pada mulanya, dan terus berkembang mengikuti perkembangan monitor desktop[2]. Hal ini disebabkan karena dalam layar yang besar, seorang desainer web dapat memasukkan banyak informasi penting dalam satu halaman. Alasan lainnya adalah komputer *desktop* dilengkapi dengan perangkat input-output yang nyaman dan memiliki kemampuan komputasi yang besar[2], sehingga gambar dan video bisa ditampilkan melalui *website* untuk *desktop*.

Akhir-akhir ini banyak pengguna yang mengakses *website* melalui perangkat *mobile*, seperti tablet atau *smartphone*. Personalisasi tampilan dan konten menjadi lebih penting. Perangkat *mobile* memiliki sifat yang berbeda

dengan komputer *desktop*. Perangkat *mobile* tidak memiliki layar berukuran besar, perangkat input-output yang nyaman, dan memiliki kemampuan komputasi terbatas[2][3][6]. Jika *website* untuk komputer *desktop* langsung ditampilkan pada perangkat *mobile*, maka informasi yang muncul pada layar perangkat *mobile* akan sulit untuk dibaca. Oleh karena itu tampilan *website* yang berbeda diperlukan agar tingkat keterbacaan informasi menjadi lebih tinggi.

Web browser menjadi salah satu aplikasi yang paling penting dalam setiap perangkat *mobile*. Hampir semua *smartphone* memiliki perangkat lunak web browser. Namun web browser default ini tidak menyediakan fitur adaptasi konten atau yang mendukung[4]. Rata rata web server yang ada hanya menyediakan dua skema layanan yang berdasarkan dari perangkat pengaksesnya. Ketika pengguna mengakses sebuah web server dari perangkat *desktop*, maka akan diberikan halaman yang biasanya ditampilkan pada perangkat desktop. Sedangkan jika pengguna mengakses melalui perangkat *mobile* seperti *smartphone* atau tablet, maka akan diberikan halaman yang disesuaikan untuk rata-rata perangkat *mobile*.

Mekanisme penyesuaian konten yang umum dilakukan oleh penyedia layanan masih

kurang maksimal. Dengan beragamnya spesifikasi dan kondisi sumberdaya perangkat *mobile* yang dinamis, maka dibutuhkan mekanisme penyesuaian konten yang dapat mendukung keberagaman perangkat dari segi sumberdaya yang dimiliki.

Dari masalah diatas, penulis telah melakukan penelitian untuk memberikan solusi dengan menyajikan hasil penelitian tentang mekanisme penyesuaian konten yang didasarkan pada informasi sumberdaya pada perangkat *mobile*. Selain itu penulis juga akan memaparkan konsep untuk pengembangan model penyesuaian konten yang lebih lanjut.

Kontribusi penulis pada penelitian ini adalah usulan untuk mengembangkan mekanisme penyesuaian konten yang mempertimbangkan kondisi sumberdaya baterai yang dinamis pada perangkat agar baterai pada perangkat *mobile* dapat bertahan lebih lama.

II. Penelitian Terkait

Penelitian yang terkait dengan penelitian yang dilakukan oleh penulis telah banyak dilakukan, namun kebanyakan dari penelitian yang telah dilakukan hanya menyediakan mekanisme adaptasi yang hanya menyediakan konsep dan meminimalisasi jumlah datayang dikirimkan ke pengguna. Dengan kata lain, konten akan dibuat seminimal mungkin agar kualitas dapat tercapai.

Penelitian yang dilakukan oleh Tian, dkk[5] memiliki fokus utama untuk mengukur performa *mobilewebsite*. Latar belakang dari penelitian yang dilakukan oleh Tian adalah penggunaan XML pada *web service* mengakibatkan ukuran web semakin besar, maka dari itu perlu digunakan kompresi pada *web service* terutama pada *clientmobile* yang cenderung memiliki konektivitas yang buruk dan biaya komunikasi yang tinggi. Kompresi merupakan salah satu cara untuk mengatasi permasalahan ukuran pesan pada *web service*. Penelitian ini juga menunjukkan bahwa kompresi sangat membantu klien yang memiliki konektivitas yang buruk serta keterbatasan perangkat yang digunakan.

Proses kompresi yang dilakukan juga memiliki beberapa keterbatasan yakni waktu CPU pada perangkat klien yang digunakan untuk kompresi dan dekompresi. Proses kompresi juga menurunkan performa CPU pada server karena adanya waktu tambahan yang digunakan untuk mengompresi dan dekompresi konten. Metode yang digunakan oleh Tian adalah dengan mengompresi data

XML *response* pada *web service*. Penelitian ini menunjukkan bahwa pada kedua ujung komputasi, baik server, maupun klien yang memiliki konektivitas yang buruk menunjukkan bahwa kompresi memberikan keuntungan melalui pendekatan yang dilakukan dengan menggunakan kompresi XML.

Sementara itu, penelitian dari Guirguis, dkk[6] memiliki fokus utama pada adaptasi konten pada *device*. Pengguna *mobile website* mengalami permasalahan umum yakni kebanyakan konten web didesain untuk pengguna komputer desktop yang memiliki ukuran layar yang besar serta memiliki koneksi internet yang cepat. Pada perangkat *mobile*, ukuran layarnya terbatas dan memiliki koneksi internet yang lebih lambat. Maka dari itu Guirguis, dkkmempertimbangkan untuk menangani batasan yang ada pada perangkat *mobile* ini.

Perangkat *mobile* memiliki kemampuan komputasi yang terbatas, memori yang lebih kecil, ukuran layar yang lebih kecil, *bandwidth* yang kecil, sumber daya yang kecil, serta daya tahan yang singkat[2][3][6]. Guirguis, dkk mengemukakan *framework* untuk *web content and resources adaptation in mobile device* (WRAMD). WRAMD dapat digunakan oleh *webmaster* sebagai alat untuk mengembangkan *mobilewebsite* dan desktop *website* dalam satu kali proses. WRAMD dapat memfasilitasi produksi konten pada website, baik *mobile* maupun desktop serta akan membuat *website* menjadi lebih ringan. Tujuan dari penelitian ini adalah untuk menyediakan tampilan *website* yang baik, meminimalisir penggunaan *bandwidth* untuk akses pada perangkat *mobile* serta menyediakan tampilan yang spesifik pada tiap perangkat *mobile*.

Adaptasi konten merupakan teknik yang penting untuk perangkat *mobile*. Sistem adaptasi konten yang telah ada telah dikembangkan dengan tujuan yang spesifik, Jiang mengemukakan sistem adaptasi konten yang dapat di kembangkan (*extensible*) yang dinamakan Xadaptor (He, J., 2007). Jiang menggunakan pendekatan berbasis *rule* untuk memfasilitasi sistem adaptasi konten yang *extensible* dan sistematis. Hal ini mengintegrasikan mekanisme adaptasi untuk beberapa tipe konten dan mengelompokkan konten-konten ini kedalam *rule-base*. *Rule* akan diatur berdasarkan informasi dari klien secara individu.

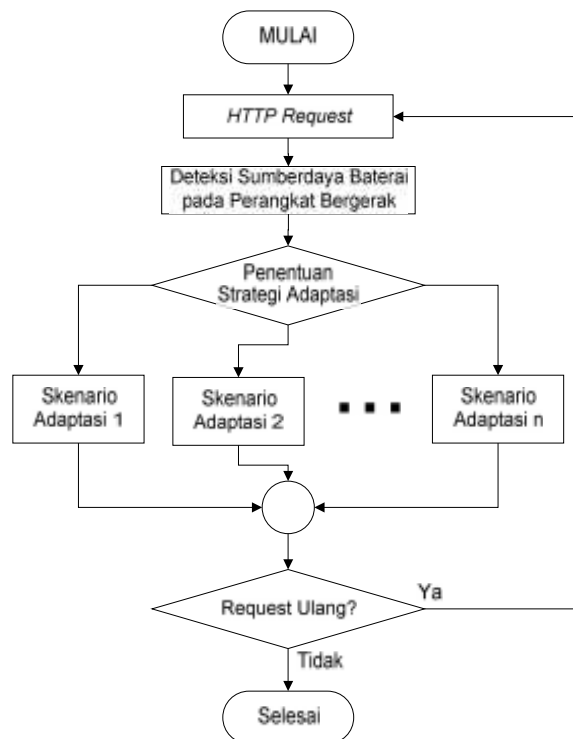
Jiang mengklasifikasikan objek pada halaman HTML menjadi struktur, konten dan penunjuk objek. Teknik adaptasi konten yang

telah ada, secara umum terfokus pada konten dan tidak mempertimbangkan teknik adaptasi untuk dokumen HTML terstruktur. Xadaptor menggunakan struktur objek pada halaman HTML untuk sistem adaptasi konten yang diterapkan.

Dengan meningkatnya keragaman akses konten pada perangkat dan teknologi jaringan, serta preferensi pengguna, pengiriman konten telah berevolusi menjadi masalah *ubiquitous service* dimana produksi, pengiriman dan akses terhadap konten membentuk tantangan baru untuk penyedia konten, operator jaringan dan user. Adaptasi konten telah diakui sebagai salah satu aspek yang penting untuk pengiriman konten *ubiquitous*. Namun hal ini masih menjadi tantangan besar tidak hanya karena kompleksitas dan keragaman teknologi, namun juga implikasi pengembangan sistem yang berdampak pada penyedia konten, operator jaringan, user, dan penyedia layanan pihak ketiga. Ding, (Ding, 2011) mengembangkan sebuah *framework* yang dinamai *Adaptation Management Framework* (AMF). AMF menargetkan distribusi konten yang merata melalui rantai pengiriman konten.

III. Desain Sistem

Sistem yang akan dibuat, digambarkan pada Gambar 1 berikut.



Gambar 1. Model Usulan Penelitian

Algoritma dimulai dengan melakukan deteksi sumberdaya pada perangkat klien seperti beban prosesor, memori dan baterai. Setelah beberapa parameter ini diketahui nilainya maka akan dilakukan proses penentuan strategi adaptasi. Proses penentuan strategi adaptasi inilah yang akan menentukan skenario adaptasi yang sesuai untuk klien.

Pada saat proses HTTP Request berlangsung, *browser* pada perangkat *mobile* akan mengirimkan informasi berupa default HTTP header yang ditunjukkan pada Gambar 2.

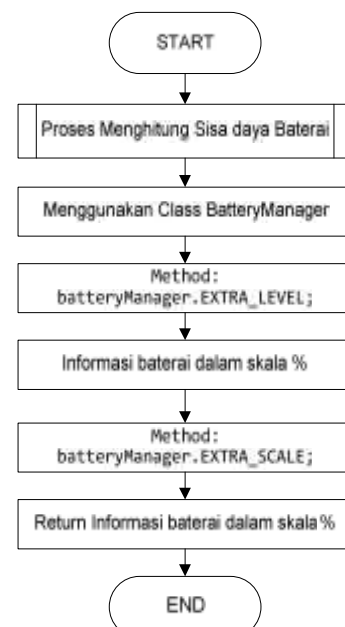
```

Get request from 202.67.46.26 on 22-10-2014
07:54:22 with header detail:
--> Host = [36.85.90.238:8080]
--> Accept-encoding = [gzip,deflate]
--> Ram-free = [21.719415]
--> Connection = [keep-alive]
--> Accept-language = [en-US]
--> User-agent = [Wee Browser]
--> Accept =
[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp.*/*;q=0.8]
  
```

Gambar 2. Default HTTP Header

A. Deteksi Sumberdaya Baterai

Deteksi sumberdaya baterai dilakukan menggunakan *class* *BatteryManager*. *Method* *batteryManager.EXTRA_LEVEL* digunakan untuk melakukan pembacaan pada kondisi baterai saat ini. *Method* *batteryManager.EXTRA_SCALE* digunakan untuk menentukan skala baterai pada perangkat *mobile*.



Gambar 5. Proses kalkulasi sumberdaya baterai

Dari dua variabel yang dihasilkan oleh *method* tersebut, kondisi baterai saat ini akan dibandingkan dengan skala baterai pada

perangkat *mobile* sehingga akan diketahui nilai sisa daya baterai dalam bentuk persentase. Diagram alir proses penghitungan sisa daya baterai ditunjukkan pada Gambar 5.

IV. Pengujian

Pengujian dilakukan untuk mengetahui apakah penelitian optimalisasi sumberdaya baterai telah tercapai. Berikut ini adalah *testbed* yang digunakan dalam penelitian ini:

Web Server

Spesifikasi *Hardware* :

- Prosesor : Intel Core2Duo T6400 2,40 GHz
- Memori : 2 GB DDR3
- Harddisk : 250 GB SATA
- NIC : TP Link Gigabit *Ethernet*
- OS : Ubuntu Server 14.04

Mobile Device

Spesifikasi *Hardware* :

- Prosesor : Intel Atom Dual Core 1,6 GHz
- Memori : 2GB
- Storage : 8 GB NAND *Flash Memory*
- WiFi : 802.11.b/g/n
- Jaringan : 2G & 3G
- OS : Android 4.3 Jelly Bean
- Browser : Android Java *Browser*

Ujicoba berdasarkan sisa daya baterai pada perangkat *mobile* dilakukan untuk mengukur kesesuaian konten yang dikirimkan oleh server dengan mempertimbangkan sisa daya baterai pada perangkat *mobile*. Jika daya baterai masih tinggi, maka akan dikirimkan konten dengan kualitas baik. Sebaliknya jika daya baterai sudah rendah, maka akan dikirimkan konten yang tidak akan menguras daya baterai dengan cepat.

Tabel 3. Rule Ujicoba Berdasarkan Sisa Daya Baterai

No	Daya Baterai	Konten	Ukuran Halaman (kB)
1	91% - 100%	Konten 1	2 MB
2	81% - 90%	Konten 2	1 MB
3	71% - 80%	Konten 3	768 kB
4	61% - 70%	Konten 4	512 kB
5	51% - 60%	Konten 5	384 kB
6	41% - 50%	Konten 6	256 kB
7	31% - 40%	Konten 7	128 kB
8	21% - 30%	Konten 8	64 kB
9	11% - 20%	Konten 9	32 kB
10	0% - 10%	Konten 10	16 kB

V. Hasil Pengujian

Hasil ujicoba berdasarkan sisa daya baterai ditunjukkan untuk mengetahui sejauh mana keberhasilan *browser* menyesuaikan layanan berupa konten yang diberikan kepada pengguna berdasarkan sisa daya baterai pada perangkat *mobile*. Gambar 4.5 Menunjukkan sisa daya baterai yang tersedia pada perangkat *mobile* sebesar 100% atau penuh ketika HTTP *request* berlangsung. Gambar 4.6 menunjukkan konten yang telah disesuaikan dengan kondisi sisa daya baterai pada perangkat *mobile*.

```
Get request from 202.67.46.26 on 22-10-2014
07:54:22 with header detail:
--> Battery-level = [100.0]
--> Cache-control = [max-age=0]
--> Host = [36.85.90.238:8080]
--> Accept-encoding = [gzip, deflate]
--> Connection = [keep-alive]
--> Accept-language = [en-US]
--> X-requested-with =
[com.example.adaptivewebbrowser]
--> User-agent = [Wee Browser]
--> Accept =
[text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8]
```

Gambar 4.5 HTTP Header Skenario Pertama

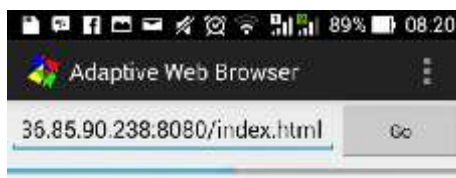


Gambar 4.6 Konten yang Dikirim oleh Server

Hasil ujicoba berdasarkan sisa daya baterai juga dilakukan untuk kondisi yang dinamis. Sebagai contoh pada skenario pengujian kedua, ditunjukkan bahwa ujicoba akses server dilakukan dengan kondisi sumberdaya baterai pada perangkat pengakses sejumlah 89% ketika HTTP *request* berlangsung. HTTP header yang menunjukkan informasi ini ditunjukkan pada Gambar 4.7 sedangkan konten yang ditunjukkan pada Gambar 4.8 juga telah disesuaikan dengan kondisi sisa daya baterai pada perangkat *mobile* yakni sejumlah 89%.

```
Get request from 202.67.46.26 on 22-10-2014 08:20:52 with header detail:
--> Battery-level = [89.0]
--> Cache-control = [max-age=0]
--> Host = [36.85.90.238:8080]
--> Accept-encoding = [gzip,deflate]
--> Connection = [keep-alive]
--> Accept-language = [en-US]
--> X-requested-with = [com.example.adaptivewebbrowser]
--> User-agent = [Wee Browser]
--> Accept = [text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8]
```

Gambar 4.7 HTTP Header Skenario Kedua



**Institut Teknologi
Sepuluh Nopember**

Profil Kampus



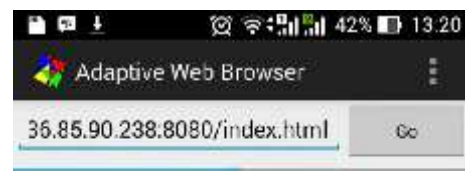
Gambar 4.8 Konten yang Dikirim oleh Server

Hasil ujicoba berdasarkan sisa daya baterai juga dilakukan untuk kondisi yang berbeda. Pada skenario pengujian ketiga, ditunjukkan

bahwa ujicoba akses server dilakukan dengan kondisi sumberdaya baterai pada perangkat pengakses menunjukkan nilai 42% ketika HTTP *request* dilakukan. HTTP header pada skenario ini ditunjukkan pada Gambar 4.9. Konten yang telah disesuaikan dengan kondisi sisa daya baterai pada perangkat *mobile* yakni sejumlah 42% ditunjukkan pada Gambar 4.10.

```
Get request from 202.67.46.26 on 22-10-2014 13:20:52 with header detail:
--> Battery-level = [42.0]
--> Cache-control = [max-age=0]
--> Host = [36.85.90.238:8080]
--> Accept-encoding = [gzip,deflate]
--> Connection = [keep-alive]
--> Accept-language = [en-US]
--> X-requested-with = [com.example.adaptivewebbrowser]
--> User-agent = [Wee Browser]
--> Accept = [text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8]
```

Gambar 4.9 HTTP Header Skenario Ketiga



**Institut Teknologi
Sepuluh Nopember**

Profil Kampus



Institut Teknologi Sepuluh Nopember (ITS) adalah sebuah perguruan tinggi negeri yang terletak di Surabaya. ITS awalnya didirikan oleh Yayasan Perguruan Tinggi Teknik (YPTT) yang diketuai oleh dr. Angka Nitisastra pada tanggal 10 November 1957. Nama ITS mulai digunakan melalui Peraturan Pemerintah No. 9 tahun 1961 (ditetapkan tanggal 23 Maret 1961) yang juga menetapkan bahwa ITS menjadi ITS.

Gambar 4.10 Konten yang Dikirim oleh Server

Dari hasil pengujian yang dilakukan didapatkan bahwa penelitian tentang *browser* yang adaptif terhadap informasi sumberdaya pada perangkat pengakses telah berhasil dilakukan sesuai dengan *rule* yang dirancang. Hal ini akan memberikan sebuah alternatif baru bagi para pengembang aplikasi web untuk

melakukan adaptasi konten sesuai dengan kondisi sumberdaya pada perangkat pengakses. Jika hal ini diterapkan, maka akan terjadi peningkatan daya tahan baterai pada perangkat pengakses.

VI. Kesimpulan dan Saran

Di era modern saat ini, teknologi dan perangkat *mobile* telah menjadi kebutuhan primer untuk mengakses informasi. Lebih dari 80% pengakses internet menggunakan perangkat *mobile*. Keterbatasan akan daya baterai pada perangkat *mobile* juga menjadi salah satu kendala pengguna untuk mengakses informasi dalam waktu yang lama. Dengan penelitian ini, diharapkan daya baterai akan bertahan lebih lama karena informasi yang didapatkan bisa disesuaikan dengan kondisi sumberdaya baterai pada perangkat.

Saran pengembangan selanjutnya adalah untuk menyisipkan informasi lain pada HTTP header yang dapat mendukung kualitas pengiriman konten, baik dari segi waktu respon, kecepatan pemrosesan, beban memori dan sebagainya.

REFERENSI

- [1]. Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018"
[Online]. http://www.cisco.com/c/en/us/solution/s/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.
Diakses tanggal 1 September 2014.
- [2]. B'Far, R., "Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML", Cambridge, 2004.
- [3]. Ojala, O., "Service Oriented Architecture in Mobile Devices: Protocols and Tools", Helsinki University of Technology, 2005
- [4]. Wang, H., Kong, J., Guo, Y., Chen, X., "Mobile Web Browser Optimizations in the Cloud Era: A Survey" 2013 IEEE 7th International Symposium on Service Oriented System Engineering (SOSE), pp 527 – 536, 2013
- [5]. Tian, M., Gramm, A., Ritter, H., Schiller, J.H., "Adaptive QoS for Mobile Web Services through Cross-layer Communication", Computer Section Volume:40 , Issue: 2 pp 59-63, 2007
- [6]. Guirguis, S.K., Hassan, M.A., "A Smart Framework For Web Content And Resources Adaptation In Mobile Devices", The 12th International Conference on Advanced Communication Technology (ICACT) Volume 1 pp 487-492, 2010
- [7]. Jiang He; Tong Gao; Wei Hao; I-Ling Yen; Bastani, F., "A Flexible Content Adaptation System Using a Rule-Based Approach," Knowledge and Data Engineering, IEEE Transactions on , vol.19, no.1, pp.127,140, Jan. 2007
- [8]. Jie Ding; Ning Li, "A Distributed Adaptation Management Framework in Content Delivery Networks," Wireless Communications, Networking and Mobile Computing (WiCOM), 2011 7th International Conference on , vol., no., pp.1,4, 23-25 Sept. 2011